php

Search

PHP 5.6.0beta2 released

FTP

Introduction
Installation/Configuration
Constantes pré-définies
Exemples
» **Fonctions FTP**

Change language: French

Edit Report a Bug

Fonctions FTP

Sommaire

- ftp_alloc — Alloue de l'espace pour un téléchargement de fichier
- ftp_cdup — Change de dossier et passe au dossier parent
- ftp_chdir — Modifie le dossier FTP courant
- ftp_chmod — Modifie les droits d'un fichier via FTP
- ftp_close — Ferme une connexion FTP
- ftp_connect — Ouvre une connexion FTP
- ftp_delete — Efface un fichier sur un serveur FTP
- ftp_exec — Exécute une commande sur un serveur FTP
- ftp_fget — Télécharge un fichier via FTP dans un fichier local
- ftp_fput — Charge un fichier sur un serveur FTP
- ftp_get_option — Lit différentes options pour la connexion FTP courante
- ftp_get — Télécharge un fichier depuis un serveur FTP
- ftp_login — Identification sur un serveur FTP
- ftp_mdtm — Retourne la date de dernière modification d'un fichier sur un serveur FTP
- ftp_mkdir — Crée un dossier sur un serveur FTP
- ftp_nb_continue — Reprend le téléchargement d'un fichier (non bloquant)
- ftp_nb_fget — Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)
- ftp_nb_fput — Écrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant)
- ftp_nb_get — Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)
- ftp_nb_put — Envoie un fichier sur un serveur FTP (non-bloquant)
- ftp_nlist — Retourne la liste des fichiers d'un dossier
- ftp_pasv — Active ou désactive le mode passif
- ftp_put — Charge un fichier sur un serveur FTP
- ftp_pwd — Retourne le nom du dossier courant
- ftp_quit — Alias de ftp_close
- ftp_raw — Envoie une commande FTP brute
- ftp_rawlist — Fait une liste détaillée des fichiers d'un dossier
- ftp_rename — Renomme un fichier sur un serveur FTP
- ftp_rmdir — Efface un dossier FTP
- ftp_set_option — Modifie les options de la connexion FTP
- ftp_site — Exécute la commande SITE sur un serveur FTP
- ftp_size — Retourne la taille d'un fichier
- ftp_ssl_connect — Ouvre une connexion FTP sécurisée avec SSL
- ftp_systype — Retourne un identifiant de type de serveur FTP

User Contributed Notes 16 notes

1 ras "at" fyn {dot} dk 7 years ago

A function to get a remote file and return it's contents, instead of saving to a local file, was missing - here it is:

```
function ftp_fetch($ftp_stream, $remote_file) {
   ob_end_flush();
   ob_start();
   $out = fopen('php://output', 'w');
   if (!ftp_fget($ftp_stream, $out, $remote_file, FTP_ASCII)) die('Unable to get file: ' . $remote_file);
   fclose($out);
   $data = ob_get_clean();
   return $data;
}
```

It works the same as ftp_get(), but instead returns the contents of the remote file - for example:

```
$ftp = ftp_connect('my.server.com', 21, 60);
ftp_login($ftp, 'username', 'password');
$data = ftp_fetch($ftp, 'path/to/remote.file');
echo $data;
```

Note, I use it to fetch text-files from a server - if you need to fetch binary files, change FTP_ASCII to FTP_BINARY .. but most likely, getting files to memory is only useful for smaller files, e.g. plain text, xml, etc.

0 spam at kacke dot de 6 years ago

Here is a FTP-abstraction-class which supports the most commands.

Also it works on TLS, SSLv2 etc. by using stream_crypto_enable()

http://kacke.de/php_samples/source.php?f=ftp.cls.php

Many thanks to WEZ at this point after all the years :-)

Toppi

0 dot a dot@dot dot com 6 years ago

This took me a few hours to work out. It is based on the code below I've just added a string replace to convert pathname given by Windows to Unix pathname. Hope it saves somebody some time. Enjoy! (Also, delete file function is included:))

```
<?php
$myFile = $_FILES['file']; // This will make an array out of the file information that was stored.
   $file = $myFile['tmp_name'];  //Converts the array into a new string containing the path name on the server where your file is.
   $myFileName = $_POST['MyFile']; //Retrieve file path and file name
   $myfile_replace = str_replace('\\', '/', $myFileName);    //convert path for use with unix
   $myfile = basename($myfile_replace);    //extract file name from path
   $destination_file = "/".$myfile;  //where you want to throw the file on the webserver (relative to your login dir)
   // connection settings
   $ftp_server = "127.0.0.1";  //address of ftp server (leave out ftp://)
   $ftp_user_name = ""; // Username
   $ftp_user_pass = "";   // Password
   $conn_id = ftp_connect($ftp_server);        // set up basic connection
   // login with username and password, or give invalid user message
   $login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass) or die("<h1>You do not have access to this ftp server!</h1>");
   $upload = ftp_put($conn_id, $destination_file, $file, FTP_BINARY);  // upload the file
```

```
    if (!$upload) {  // check upload status
        echo "<h2>FTP upload of $myFileName has failed!</h2> <br />";
    }
/*
    // try to delete $file
    if (ftp_delete($conn_id, $destination_file)) {
        echo "$destination_file has been deleted!\n";
    } else {
        echo "Could not delete $destination_file!\n";
    }
*/
ftp_close($conn_id); // close the FTP stream
}
?>
<form name="form" enctype="multipart/form-data" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST" >
Please choose a file: <input type="file" name="file" accept="text/plain" onChange="MyFile.value=file.value">
<input name="MyFile" type="hidden" id="MyFile" tabindex="99" size="1" />
<input type="submit" name="submit" value="upload" style="vertical-align:middle"/><br/><br/>
</form>
```

0 breese at pipetech dot tv 7 years ago

I fixed the below upload script to work with Windows uploads (and updated some of the syntax). I also added a Javascript "progress indicator" for those of us who don't use PHP 5.2

```
<?php
if(isset($_POST['start_upload']) && $_FILES['txt_file']['name'] != ""){

    $local_file = $_FILES['txt_file']['tmp_name']; // Defines Name of Local File to be Uploaded

    $destination_file = "/".basename($_FILES['txt_file']['name']);  // Path for File Upload (relative to your login dir)

    // Global Connection Settings
    $ftp_server = "127.0.0.1";      // FTP Server Address (exlucde ftp://)
    $ftp_user_name = "username";     // FTP Server Username
    $ftp_user_pass = "password";      // Password

    // Connect to FTP Server
    $conn_id = ftp_connect($ftp_server);
    // Login to FTP Server
    $login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

    // Verify Log In Status
    if ((!$conn_id) || (!$login_result)) {
        echo "FTP connection has failed! <br />";
        echo "Attempted to connect to $ftp_server for user $ftp_user_name";
        exit;
    } else {
        echo "Connected to $ftp_server, for user $ftp_user_name <br />";
    }

    $upload = ftp_put($conn_id, $destination_file, $local_file, FTP_BINARY);  // Upload the File

    // Verify Upload Status
    if (!$upload) {
        echo "<h2>FTP upload of ".$_FILES['txt_file']['name']." has failed!</h2><br /><br />";
    } else {
        echo "Success!<br />" . $_FILES['txt_file']['name'] . " has been uploaded to " . $ftp_server . $destination_file . "!<br /><br />";
```

```
    }

    ftp_close($conn_id); // Close the FTP Connection
}
?>

<html>
    <head>
        <script type="text/javascript">
            window.onload = function() {
                document.getElementById("progress").style.visibility = "hidden";
                document.getElementById("prog_text").style.visibility = "hidden";
            }

            function dispProgress() {
                document.getElementById("progress").style.visibility = "visible";
                document.getElementById("prog_text").style.visibility = "visible";
            }

        </script>

    </head>
    <body>
        <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="POST" enctype="multipart/form-data">
            Please choose a file: <input name="txt_file" type="file" size="35" />
            <input type="submit" name="start_upload" value="Upload File" onClick="dispProgress()" />
        </form>

        <!-- Link to progress file: see http://www.ajaxload.info/ for animated gifs -->
        <img id="progress" src="http://www.your.site/images/progress.gif" />
        <p id="prog_text" style="display:inline;"> Upload Started!</p>

    </body>
<html>
```

0 cgray at premierconsulting dot com (Chris Gray) 7 years ago

After looking everywhere for a "FULLY WORKING" ftp uploader script, I finally came up with this, by using a lot of various one's I found on the web. If you find a better way of doing something, please let me know. Thanks, and Enjoy

```
<?php
 if(isset($_POST['SubmitFile'])){
    $myFile = $_FILES['txt_file']; // This will make an array out of the file information that was stored.
    $file = $myFile['tmp_name']; //Converts the array into a new string containing the path name on the server where your file is.

    $myFileName = basename($_POST['txt_fileName']); //Retrieve filename out of file path

    $destination_file = "/".$myFileName; //where you want to throw the file on the webserver (relative to your login dir)

    // connection settings
    $ftp_server = "127.0.0.1"; //address of ftp server.
    $ftp_user_name = "Your UserName"; // Username
    $ftp_user_pass = "Your Password"; // Password

    $conn_id = ftp_connect($ftp_server);      // set up basic connection
    $login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass) or die("<h2>You do not have access to this ftp server!</h2>");
// login with username and password, or give invalid user message
    if ((!$conn_id) || (!$login_result)) {  // check connection
```

```
        // wont ever hit this, b/c of the die call on ftp_login
        echo "FTP connection has failed! <br />";
        echo "Attempted to connect to $ftp_server for user $ftp_user_name";
        exit;
      } else {
      //    echo "Connected to $ftp_server, for user $ftp_user_name <br />";
    }

    $upload = ftp_put($conn_id, $destination_file, $file, FTP_BINARY);  // upload the file
    if (!$upload) {  // check upload status
      echo "<h2>FTP upload of $myFileName has failed!</h2> <br />";
    } else {
      echo "Uploading $myFileName Complete!<br /><br />";
    }

    ftp_close($conn_id); // close the FTP stream
  }
?>

<html>
  <head>

  </head>
  <body>
      <form enctype="multipart/form-data" action="<?php echo $_SERVER['PHP_SELF'];?>" method="POST">
        Please choose a file: <input name="txt_file" type="file" id="txt_file" tabindex="1" size="35"
onChange="txt_fileName.value=txt_file.value" />
        <input name="txt_fileName" type="hidden" id="txt_fileName" tabindex="99" size="1" />

        <input type="submit" name="SubmitFile" value="Upload File" accesskey="ENTER" tabindex="2" />
    </form>
  </body>
<html>
```

0 nicklas at programmera dot nu 7 years ago

I didn't quite like my FTP program since I had to make all the settings all over when i restarted the client, so I made this script for simple ftp uploading, its running as a CLI app.

```
<?php

  //if STDIN isn't defined, define it
  if (!defined("STDIN"))
  {

    define("STDIN", fopen("php://stdin", "r"));

  }

  echo "FTP file uploader\r\n\r\n";

  echo "Server: ";

  //trim() all the fgets cause it seams to add a \n
  $server = trim(fgets(STDIN));

  echo "Username: ";
```

```
$username = trim(fgets(STDIN));

echo "Password: ";

$password = trim(fgets(STDIN));

$connect = ftp_connect($server);
$login = ftp_login($connect, $username, $password);

if ((!$connect) || (!$login))
{

    exit("Login failed.\r\n");

}

echo "Connected\r\n";

echo "PASV (Y/Any key = Off): ";

if(trim(fgets(STDIN)) == "Y")
{

    //enable passive mode
    ftp_pasv($connect, true);

        echo "PASV: On\r\n";

}
else
{

    echo "PASV: Off\r\n";

}

$showContent = ftp_nlist($connect, "");

echo "Listing Contents:\r\n";

    var_dump($showContent);

echo "File to upload: ";

$file = trim(fgets(STDIN));

//escape the uploading process by typing "exit;"
if ($file == "exit;")
{

    exit;

}

//use "$file, $file" since the file shouldn't be renamed, also if its a binary file change the last argument to FTP_BINARY
$upload = ftp_put($connect, $file, $file, FTP_ASCII);

if (!$upload)
```

```
    {

        exit("Upload failed.\r\n");

    }
    else
    {

        echo "The file \"" . $file . "\" was successfully uploaded. :)\r\nSize of \"" . $file . "\": " . filesize($file) / 1024 ."kb.\r\n";
        echo "My work is done, bye.\r\n";

    }

    ftp_close($connect);

?>
```

0 magic2lantern at mail dot ru 8 years ago

This is a simple script writen by me which allows the user
run it in console and get tree from FTP Server.

```
<?php
$argv = $_SERVER["argv"];
$argv = array_slice($argv,1);
if (getenv("HTTP_HOST")) {exit("For bash execution only!");}
if ((empty($argv[0])) || (empty($argv[1]))) {exit("Usage: thisscript.php 10.0.0.10 /games/Arcade/
where 10.0.0.10 is the FTP Server and /games/Arcade is the start path on FTP\n");}

// connect to the ftp server
$connect_id = ftp_connect($argv[0]);
// login on the server
if ($connect_id) { $login_result = ftp_login($connect_id, 'anonymous', 'JustForFun'); }
// check connection
if ((!$connect_id) || (!$login_result)) {
   echo "Unable to connect the remote FTP Server!\n";
   exit;
}
function ftp_get_dir ($ftp_dir) {
global $connect_id;
$ftp_contents = ftp_nlist ($connect_id, $ftp_dir);
    for ($myCounter_for_server_elements = 0; !empty ($ftp_contents[$myCounter_for_server_elements]);
$myCounter_for_server_elements++)
    {
// get elements list in current dir
        echo $argv[0] . $ftp_dir . $ftp_contents[$myCounter_for_server_elements] . "\n";
        ftp_get_dir ($ftp_dir . $ftp_contents[$myCounter_for_server_elements] . '/');
    }
}
ftp_get_dir ($argv[1]);
// close connection
ftp_close($connect_id);
?>
```

0 leonardo at rol dot com dot ro 8 years ago

Here is an example for downloading a remote ftp structure to local server, useful for migrating sites from one server to another, especially when you are upgrading ensim ;)

http://leonardo.rol.ro/ftpleech.txt

0 dickiedyce at uk dot com 8 years ago

It may seem obvious to others, but it had me stumped for nearly an hour! If you can connect to an ftp site but some functions (list, put, get etc) don't work, then try using ftp_pasv and set passive mode on.

```
<?php

// setup $host and $file variables for your setup before here...

$hostip = gethostbyname($host);
$conn_id = ftp_connect($hostip);

// login with username and password
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

// IMPORTANT!!! turn passive mode on
ftp_pasv ( $conn_id, true );

if ((!$conn_id) || (!$login_result)) {
  echo "FTP connection has failed!";
  echo "Attempted to connect to $host for user $ftp_user_name";
  die;
} else {
  echo "Connected to $host, for user $ftp_user_name<br>";
  echo "Host IP is $hostip<br>";

  // upload a file
  if (ftp_put($conn_id, $remote_file, $file, FTP_ASCII)) {
    echo "successfully uploaded $file<br>";
  } else {
    echo "There was a problem while uploading $file<br>";
  }

  // close the connection
  ftp_close($conn_id);
}
?>
```

0 arjen at queek dot nl 9 years ago

If you prefer a OO-approach to the FTP-functions, you can use this snippet of code (PHP5 only! and does add some overhead). It's just a "start-up", extend/improve as you wish...
You can pass all ftp_* functions to your object and stripping ftp_ of the function name. Plus, you don't have to pass the ftp-resource as the first argument.

For example:

```
<?php
ftp_delete($ftp, $file);    // Where $ftp is your ftp-resource
?>
```

Can become:

```
<?php
$ftp->delete($file);       // Where $ftp is your FTP-object
?>
```

Code:

```
<?php

class FTP {

   private $ftp;

   /* public Void __construct(): Constructor */
   public function __construct($host, $port = 21, $timeout = 90) {
      $this->ftp = ftp_connect($host, $port, $timeout);
   }

   /* public Void __destruct(): Destructor */
   public function __destruct() {
      @ftp_close($this->ftp);
   }

   /* public Mixed __call(): Re-route all function calls to the PHP-functions */
   public function __call($function, $arguments) {
      // Prepend the ftp resource to the arguments array
      array_unshift($arguments, $this->ftp);

      // Call the PHP function
      return call_user_func_array('ftp_' . $function, $arguments);
   }

}

?>
```

0 Kristy Christie (kristy at isp7 dot net) 9 years ago

Here's a little function that I created to recurse through a local directory and upload the entire contents to a remote FTP server.

In the example, I'm trying to copy the entire "iwm" directory located at /home/kristy/scripts/iwm to a remote server's /public_html/test/ via FTP.

The only trouble is that for the line "if (!ftp_chdir($ftpc,$ftproot.$srcrela))", which I use to check if the directory already exists on the remote server, spits out a warning about being unable to change to that directory if it doesn't exist.

But an error handler should take care of it.

My thanks to the person who posted the snippet on retrieving the list of files in a directory.

For the version of the script that echo's it's progress as it recurses & uploads, go to: http://pastebin.com/73784

```
<?php

// ------------------------------------------------------------------
// THE TRIGGER
// ------------------------------------------------------------------

   // set the various variables
   $ftproot = "/public_html/test/";
   $srcroot = "/home/kristy/scripts/";
   $srcrela = "iwm/";

   // connect to the destination FTP & enter appropriate directories both locally and remotely
```

```
    $ftpc = ftp_connect("ftp.mydomain.com");
    $ftpr = ftp_login($ftpc,"username","password");

    if ((!$ftpc) || (!$ftpr)) { echo "FTP connection not established!"; die(); }
    if (!chdir($srcroot)) { echo "Could not enter local source root directory."; die(); }
    if (!ftp_chdir($ftpc,$ftproot)) { echo "Could not enter FTP root directory."; die(); }

    // start ftp'ing over the directory recursively
    ftpRec ($srcrela);

    // close the FTP connection
    ftp_close($ftpc);

// ---------------------------------------------------------------------
// THE ACTUAL FUNCTION
// ---------------------------------------------------------------------
function ftpRec ($srcrela)
{
    global $srcroot;
    global $ftproot;
    global $ftpc;
    global $ftpr;

    // enter the local directory to be recursed through
    chdir($srcroot.$srcrela);

    // check if the directory exists & change to it on the destination
    if (!ftp_chdir($ftpc,$ftproot.$srcrela))
    {
        // remote directory doesn't exist so create & enter it
          ftp_mkdir    ($ftpc,$ftproot.$srcrela);
          ftp_chdir    ($ftpc,$ftproot.$srcrela);
    }

    if ($handle = opendir("."))
    {
       while (false !== ($fil = readdir($handle)))
       {
          if ($fil != "." && $fil != "..")
          {
             // check if it's a file or directory
             if (!is_dir($fil))
             {
                // it's a file so upload it
                ftp_put($ftpc, $ftproot.$srcrela.$fil, $fil, FTP_BINARY);
              }
              else
              {
                 // it's a directory so recurse through it
                 if ($fil == "templates")
                 {
                        // I want the script to ignore any directories named "templates"
                        // and therefore, not recurse through them and upload their contents
                 }
                 else
                 {
                    ftpRec ($srcrela.$fil."/");
                    chdir ("../");
```

```
            }
          }
        }
      }
      closedir($handle);
    }
  }
?>
```

0 postmaster at alishomepage dot com 10 years ago

I have written an OpenSource ZIP2FTP interface, which actually takes a given ZIP file and decompresses it in the folder on an FTP server you specify...

Therefore it may be quite interesting for you people interested in FTP, its adress is http://zip2ftp.alishomepage.com ; those who directly want the source may visit http://zip2ftp.alishomepage.com/?do=getSource

0 Vikrant Korde <vakorde at hotmail dot com> 10 years ago

```
<?

//If you want to move or replicate the folder hierarchy from your current server to another remote server. Then this will be helpful as this will browse the current server's directory and at the same time it will copy that file in the remote server in the same directory.

//This script will copy all the files from this directory and subdirectory to another remote server via FTP

function rec_copy ($source_path, $destination_path, $con)
{
   ftp_mkdir($con, $destination_path);
   ftp_site($con, 'CHMOD 0777 '.$destination_path);
   ftp_chdir($con,$destination_path);

   if (is_dir($source_path))
   {
      chdir($source_path);
      $handle=opendir('.');
      while (($file = readdir($handle))!==false)
      {
         if (($file != ".") && ($file != ".."))
         {
            if (is_dir($file))
            {
               // here i am restricting the folder name 'propertyimages' from being copied to remote server. -- VK
               if($file != "propertyimages")
               {
                  rec_copy ($source_path."/".$file, $file, $con);
                  chdir($source_path);
                  ftp_cdup($con);
               }
            }
            if (is_file($file))
            {
               $fp = fopen($file,"r");
               // this will convert spaces to '_' so that it will not throw error.  -- VK
               ftp_fput ($con, str_replace(" ", "_", $file), $fp,FTP_BINARY);
               ftp_site($con, 'CHMOD 0755 '.str_replace(" ", "_", $file));
            }
         }
```

```
        }
        closedir($handle);
    }
}

// make a FTP connection --VK
$con = ftp_connect("69.18.213.131",21);
$login_result = ftp_login($con,"username","password");

// this is the root path for the remote server-- VK
$rootpath = "mainwebsite_html";

// this is the physical path of the source directory. actually u can also use the relative path. -- VK
$sourcepath = realpath("../")."/resdesk";

// this directory name will only change the top most directory and not the inner one -- VK
$destination_dir_name = "resdesk_".$account_id."/";

rec_copy ($sourcepath, $destination_dir_name, $con);
if (function_exists("ftp_close"))
{
    ftp_close($con);
}
?>
```

0 NOSPAMkent at ioflux dot NOSPAM dot com 11 years ago

I think what some other posts were trying to say which may need clarification is that in PHP 4.2.3, ftp_connect("myhost.com") was failing most of the time, except it would work like every few minutes.

The fix is that ftp_connect seems to have a bug resolving addresses. If you do:

```
$hostip = gethostbyname($host);
$conn_id = ftp_connect($hostip);
```

It seems to solve the problem.
(Other users referred to an ftpbuf() error... not sure what that is, but this should fix it.)

0 sven at cartell-network dot de 12 years ago

connection to a ftp server across proxy

```
$ftp_server = "proxy"; f.e. 123.456.789.10
$ftp_user_name = "username@ftpserver"; f.e. exampleuk@www.example.uk
$ftp_user_pass = "password";

$conn_id = ftp_connect($ftp_server, 2121);
$login_result = ftp_login( $conn_id, $ftp_user_name, $ftp_user_pass );
```

-1 postmaster at alishomepage dot com 10 years ago

Here's another FTP interface over PHP (also uses MySQL)

http://myftp.alishomepage.com

PS: this script will ALSO allow you to download its source... So it becomes interesting for YOU PROGRAMMERS as well :D

add a note

Copyright © 2001-2014 The PHP Group
My PHP.net
Contact
Other PHP.net sites
Mirror sites
Privacy policy